Alzheimer's Disease Detection

&

Interpretability Enhancement Using LRP.

Report Submitted in partial fulfillment of requirements for the B-Tech Degree in Computer Science and Engineering.

NAME OF THE STUDENT	ROLL NUMBER
Aakash Kumar	2020UCA1865
Mohd Farman	2020UCA1878
Naman Singla	2020UCA1882

Under the supervision of Prof. Ritu Sibbal.

Department of Computer Science and Engineering

(Netaji Subhash University of Technology)

New Delhi, India-110078

<u>CERTIFICATE</u>

Department of Computer Science and Engineering.

This is to certify that the work embodied in this project thesis titled, **"Alzheimer Disease Detection & Interpretability using LRP"** by Aakash Kumar (2020UCA1865), Mohd Farman (2020UCA1878) and Naman Singla (2020UCA1882) is the bona fide work of the group submitted to Netaji Subhash University of Technology for consideration in 7th Semester B-tech project Evaluation.

The original research work was carried out under my/our guidance and supervision in the academic year 2023-2024. This work has not been submitted for any other degree or diploma of any university. On the basis of a declaration made by the group, we recommend the project report for evaluation.

Prof. Ritu Sibbal

CANDIDATE(S) DECLARATION

Department of Computer Science and Engineering

I/We Aakash Kumar(2020UCA1865), Mohd Farman(2020UCA1878) and Naman Singla(2020UCA1878) of B-tech. Department of Computer Science and engineering, hereby declare that the Project-Thesis titled "**Alzheimer Disease Detection & Interpretability using LRP**" which is submitted by me/us to the department of Computer Science and Engineering, Netaji Subhas University of Technology(NSUT) Dwarka, New Delhi in partial fulfillment of the requirement for the award of the Degree of Bachelor of Technology is original and not copied from the source without proper citation. The manuscript has been subjected to plagiarism checks by Turnitin Software. This work has not previously formed the basis of the award of any degree.

Place:

Date:

Aakash Kumar	Mohd Farman	Naman Singla
(2020UCA1865)	(2020UCA1878)	(2020UCA1882)

CERTIFICATE OF DECLARATION

Department of Computer Science and Engineering.

This is to certify that the Project-Thesis called "**Alzheimer Disease Detection & Interpretability using LRP**" which is being submitted by Aakash Kumar (2020UCA1865), Mohd Farman (2020UCA1878) and Naman Singla (2020UCA1882) to the Department of Computer Science and Engineering, Netaji Subhas University of Technology (NSUT) Dwarka, New Delhi in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology, is a record of the thesis work carried out by the students under my supervision and guidance, the contents of this project work, in full or in parts, has not been submitted for any other degree or diploma.

Place:

Date:

Prof. Ritu Sibbal

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to all those who make it possible to complete this project. Special thanks to our project supervisor Prof. Ritu Sibbal whose help, stimulating suggestions and encouragement helped us in writing this report. We also sincerely thank colleagues for the time spent proofreading and correcting our mistakes.

We would like to acknowledge with much appreciation the crucial role of the staff in Computer Science and Engineering, who gave us permission to use the lab and the system and gave permission to use all necessary things related to the project.

Aakash Kumar	Mohd Farman	Naman Singla
(2020UCA1865)	(2020UCA1878)	(2020UCA1882)

Turnitin Plagiarism Report

by GROUP 138 Aakash, Farman & Naman

Submission date: 11-Dec-2023 08:05PM (UTC+0200)
Submission ID: 2255829082
File name: For_plag_check.docx (2.39M)
Word count: 3446
Character count: 20694

Turnitin Plagiarism Report

ORIGINALITY REPORT

4%	2%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.researchgate.net Internet Source	1%
2	"Inventive Systems and Control", Springer Science and Business Media LLC, 2023 Publication	1%
3	Submitted to Symbiosis International University Student Paper	<1%
4	vegibit.com Internet Source	<1%
5	Submitted to Glasgow Caledonian University Student Paper	<1%
6	Sathit Prasomphan. "Chapter 3 Ensemble Classification Technique for Cultural Heritage Image", Springer Science and Business Media LLC, 2022 Publication	<1%
7	genomemedicine.biomedcentral.com Internet Source	<1%
	link.springer.com	

8	Internet Source	<1 %
9	oaji.net Internet Source	<1 %
10	Lecture Notes in Computer Science, 2013. Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	Off		

Abstract

Alzheimer's Disease (AD) is a widespread neurodegenerative condition that requires early detection for effective intervention. This initiative utilizes artificial intelligence (AI) & deep learning methodologies to automate the classification of AD stages through the analysis of medical images. The main goal is to create a robust deep learning model that can precisely categorize various stages of Alzheimer's disease, enabling timely & targeted interventions.

Our solution uses cutting-edge deep learning techniques to evaluate medical images & shows encouraging outcomes in terms of automating the process of diagnosis & classification. The model's performance is assessed on a diverse dataset, demonstrating its potential for early & accurate identification of AD stages.

Moreover, we address the critical need for interpretability in deep learning models by incorporating Layer-wise Relevance Propagation (LRP). LRP enhances our model's transparency by attributing relevance to specific regions of the input images, providing insights into the decision-making process. This interpretability is pivotal for gaining trust in the model's predictions & aiding clinicians in understanding the underlying features influencing the classification.

The results of the project contribute to the continuous endeavors in harnessing AI for medical diagnostics, particularly within the domain of Alzheimer's Disease. The amalgamation of precise classification & improved interpretability lays the groundwork for the creation of more dependable & clinically relevant tools for the diagnosis of neurodegenerative diseases.

LIST OF CONTENTS

List of Figures

LIST OF TABLES

CHAPTER-1

INTRODUCTION

The global health challenge presented by Alzheimer's Disease (AD) necessitates innovative approaches for early diagnosis & intervention. This project strives to meet this crucial need by employing the capabilities of Artificial Intelligence (AI) & Deep Learning (DL) for classifying different stages of Alzheimer's Disease through the analysis of medical images. Acknowledging the importance of interpretability in model decisions, we incorporate Layer-wise Relevance Propagation (LRP) to improve the transparency & comprehension of our Convolutional Neural Network (CNN) model.

The foundation of our approach lies in the utilization of deep learning techniques, specifically a CNN architecture, to discern intricate patterns & features within medical images associated with different stages of Alzheimer's Disease. Leveraging the principles of transfer learning, we harness the knowledge acquired by a pre-trained neural network on a diverse dataset, allowing our model to generalize & adapt effectively to our specific AD classification task. Transfer learning not only expedites the training process but also facilitates improved performance, particularly when confronted with limited labeled medical imaging data.

A pivotal aspect of our project is the commitment to enhancing the interpretability of the CNN model's decisions. The integration of Layer-wise Relevance Propagation (LRP) enables us to unveil the black box nature of deep learning models by attributing relevance to individual pixels in the input images. This step towards interpretability is paramount for establishing trust in the model's decisions, especially in critical healthcare applications where transparency is essential for clinical acceptance.

CHAPTER-2

Motivation, Literature Survey & Problem Statement

Motivation-

Alzheimer's disease is a growing concern in healthcare, with an increasing number of cases each year. Early diagnosis allows for timely treatment & support for patients. AI can assist medical professionals in identifying Alzheimer's disease in its early stages, leading to better patient outcomes.

Our motivation is to contribute to the field of healthcare by developing an accurate Alzheimer's disease classification model. Additionally, we aim to make the model's decisions more interpretable, helping medical professionals understand the factors contributing to its predictions.

Literature Survey

In our review of the existing literature, we delved into research concerning the classification & interpretability of Alzheimer's disease. We found that this problem has been addressed using a variety of machine learning & deep learning techniques, some of which have shown encouraging results. However, a common limitation is the lack of interpretability in many existing models, posing a challenge for medical professionals to trust & effectively utilize these models. Identifying this gap, we determined that methods that improve interpretability—like Layer-wise Relevance Propagation (LRP)—are essential for offering insightful information about the model's decision-making process. LRP, as a technique, offers a way to understand the contribution of individual input pixels to the network's output by assigning relevance scores to each pixel. This process highlights the regions that play a crucial role in the classification decision, making LRP a valuable tool for interpreting deep learning models & improving their explainability.

Generic rule for implementing LRP Efficiently-

$$R_j = \sum_k \frac{A_j.\rho(W_{jk})}{\epsilon+\sum_{0,j} A_j.\rho(W_{jk})} R_k, \qquad \textbf{Eq. (1)}$$

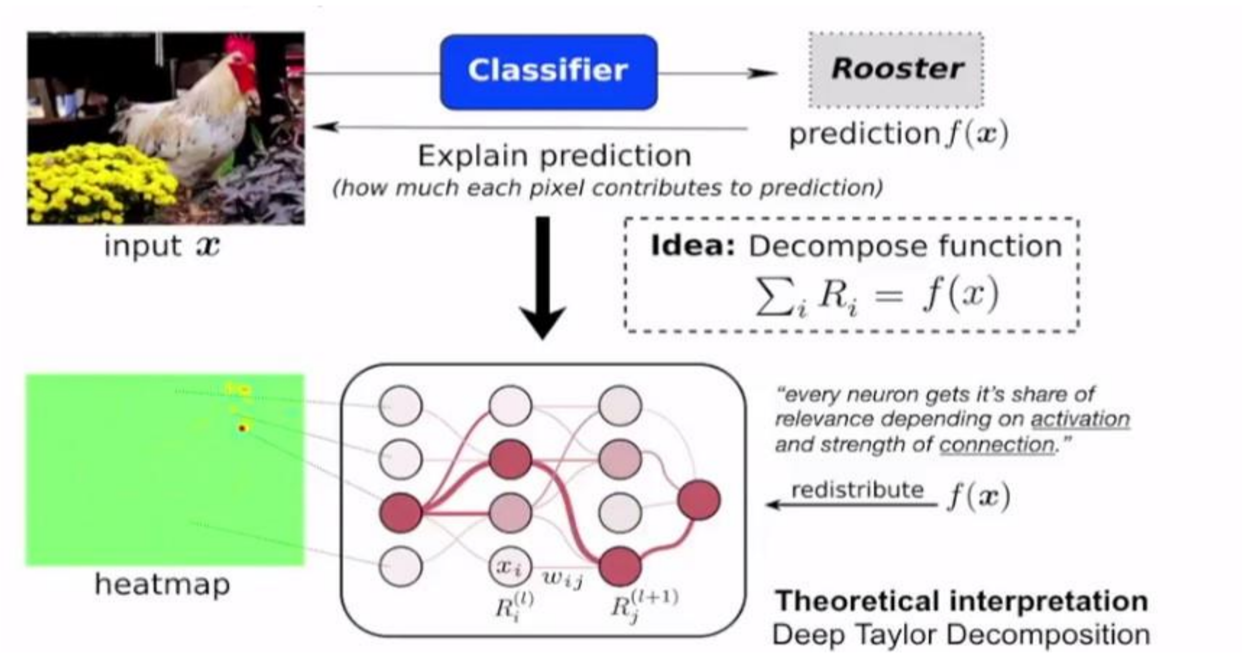
The calculation of this propagation rule can be broken down into four distinct steps:

$\forall_k: Z_k = \epsilon + \sum_{0,j} A_j.\rho(W_{jk})$ **(FORWARD PASS)** **Eq. (2)**

$\forall_k: S_k = R_k/Z_k$ **(ELEMENT WISE DIVISION)** **Eq. (3)**

$\forall_j: C_j = \sum_k \rho(W_{jk}).S_k$ **(BACKWARD PASS)** **Eq. (4)**

$\forall_j: R_j = A_j C_j$ **(ELEMENT WISE PRODUCT)** **Eq. (5)**



(i) **Working of LRP**

Problem Statement

Alzheimer's Disease poses a significant global health challenge, demanding precise & timely diagnostic solutions. Traditional methods are often labor-intensive & prone to subjectivity. In this context, the project addresses the need for an efficient, automated classification system using Artificial Intelligence & Deep Learning on medical images. The primary challenge lies in developing a robust Convolutional Neural Network (CNN) model, incorporating transfer learning for improved performance, & concurrently ensuring interpretability through Layer-wise Relevance Propagation (LRP). This endeavor seeks to overcome the limitations of current diagnostic approaches, providing a more accurate & transparent tool for early identification & classification of Alzheimer's Disease stages.

Chapter-3

OBJECTIVE AND METHODOLOGY

OBJECTIVE-

The primary objectives of our project are as follows:

Create a deep learning model dedicated to classifying Alzheimer's disease. Improve the interpretability of the model's predictions through the integration of Layer-wise Relevance Propagation (LRP).

Contribute to advancing the comprehension of Alzheimer’s disease classification & interpretability within the domain of medical image analysis.

METHODOLOGY-

Data collection & preprocessing

We collected a medical dataset related to Alzheimer's disease classification from Kaggle, consisting of brain images. The dataset includes images from different disease stages, providing valuable training & testing data.

Data preprocessing involved handling missing values, removing duplicates, label encoding, data augmentation, & data normalization. These steps were essential for preparing the data for training.

Selection of VGG16 Model

The VGG16 architecture was chosen as the foundational model for our study because of its shown efficacy in picture classification challenges. We replaced the last layer of VGG16 to adapt it to our specific problem of Alzheimer's disease classification.

Model Modifications for Alzheimer's Classification

To tailor VGG16 to our classification task, we modified the output layer to have four neurons corresponding to the four disease stages. This customization allowed our model to output disease stage predictions.

Training Process

We trained the modified VGG16 model on the preprocessed data using the Adam optimizer & a learning rate scheduler. Cross-entropy loss was used as the optimization criterion. The model underwent training over several epochs to improve its performance.

Evaluation Metrics

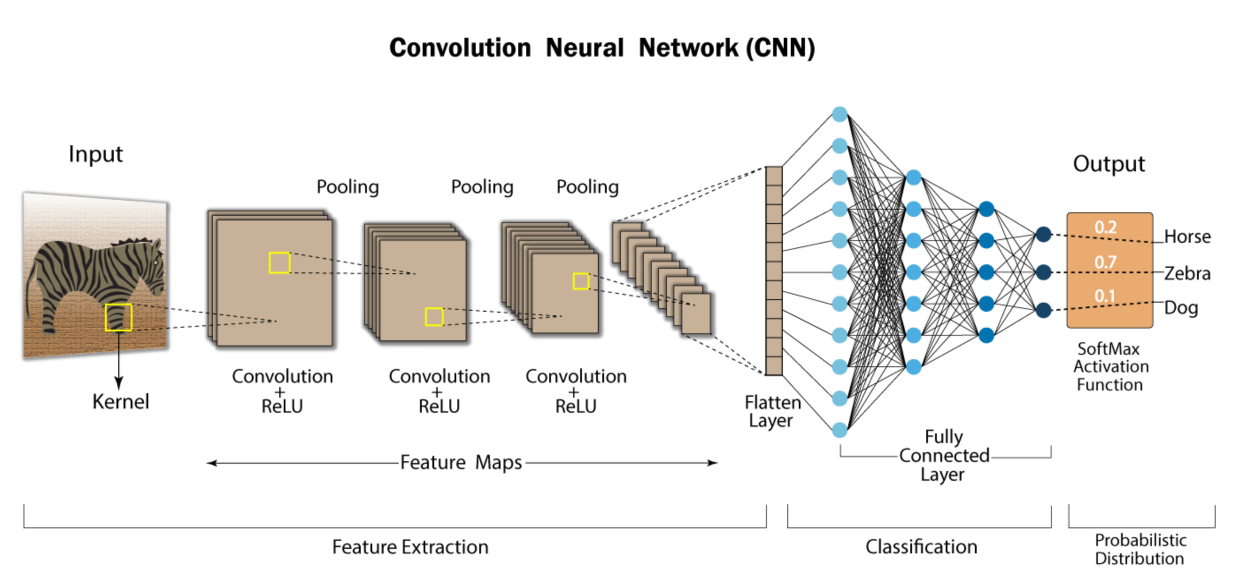
We tested the model on a different test dataset in order to gauge its effectiveness. Accuracy, precision, recall, F1-score, & execution time were among the evaluation metrics.

Convolution Neural Networks

CNNs are a particular kind of neural network used for visual data processing & analysis. When it comes to computer vision tasks like object detection, segmentation, image & video recognition, & image classification, CNNs are especially effective. Convolutional layers are the main tool used by CNNs to automatically & adaptively learn hierarchical representations of the input data. These networks are well-suited for tasks where the spatial arrangement of features is important, as in images. [4]

The ability of CNNs to automatically learn hierarchical representations from input data makes them particularly effective in extracting meaningful features. While constructing CNN, we use many types of layers stacked sequentially where each layer has their own use. Each layer is responsible for learning specific features of the input data in a hierarchical manner.

Convolutional layers use filters to detect low-level features like edges, textures, & simple patterns. Higher-level features & intricate patterns are discovered as we delve further into the network. Deep CNN designs have produced state-of-the-art results in image classification competitions such as ImageNet, where CNNs have demonstrated remarkable success in a variety of computer vision applications.



(ii) <u>CNN Architecture</u>

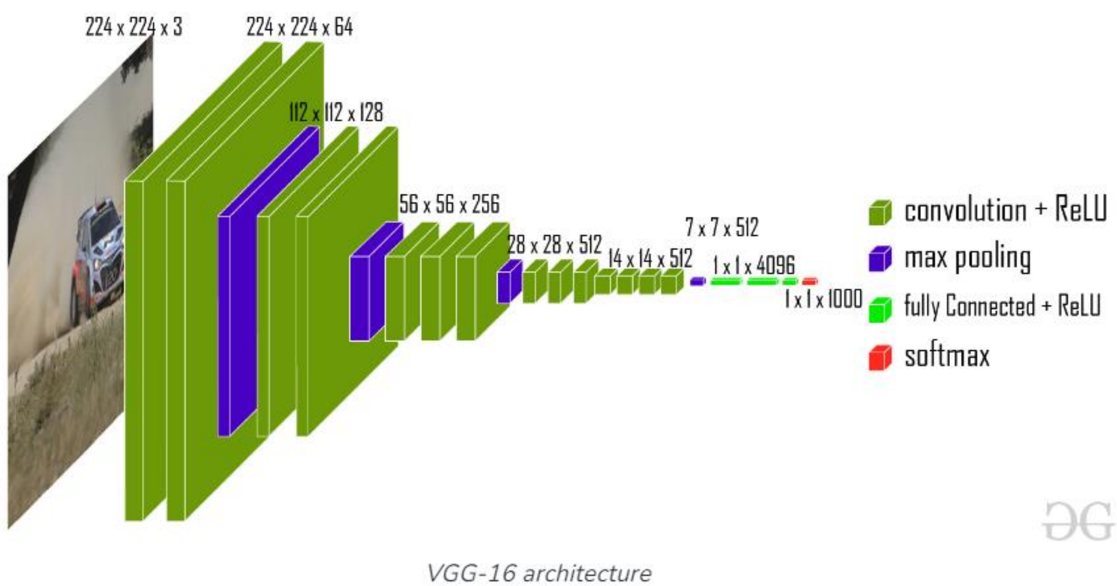
VGG16 Model-

The convolutional neural network (CNN) architecture known as VGG16, or Visual Geometry Group 16, is intended for use in image recognition & classification applications. The University of Oxford's Visual Geometry Group first presented it, & it became well-known after it took part in the 2014 ImageNet Large Scale Visual Recognition Challenge.

The architecture of the VGG16 model is marked by its homogeneity & simplicity. There are three fully connected layers & thirteen convolutional layers among its sixteen weight layers. The model has a deep architecture since the convolutional layers are stacked one after the other & have tiny 3x3 filters attached to them. By using these tiny filters, the network may learn more intricate features with comparatively few parameters.

The design is composed of five groups, each of which has a max-pooling layer after one or more convolutional layers. In order to lower the computational load & manage overfitting, the max-pooling layers are utilized to decrease the spatial dimensions of the input volume. The final classification output is produced by the fully connected layers at the end of the network, which analyze the high-level information that the convolutional layers extracted.

VGG16 has demonstrated impressive performance in various image classification tasks & serves as a benchmark for deep learning models. Despite its simplicity, it remains widely used & has paved the way for more complex architecture. Researchers & practitioners often use pre-trained versions of VGG16 for transfer learning, leveraging its learned features for tasks beyond image classification.



(iii) <u>VGG16 Architecture</u>

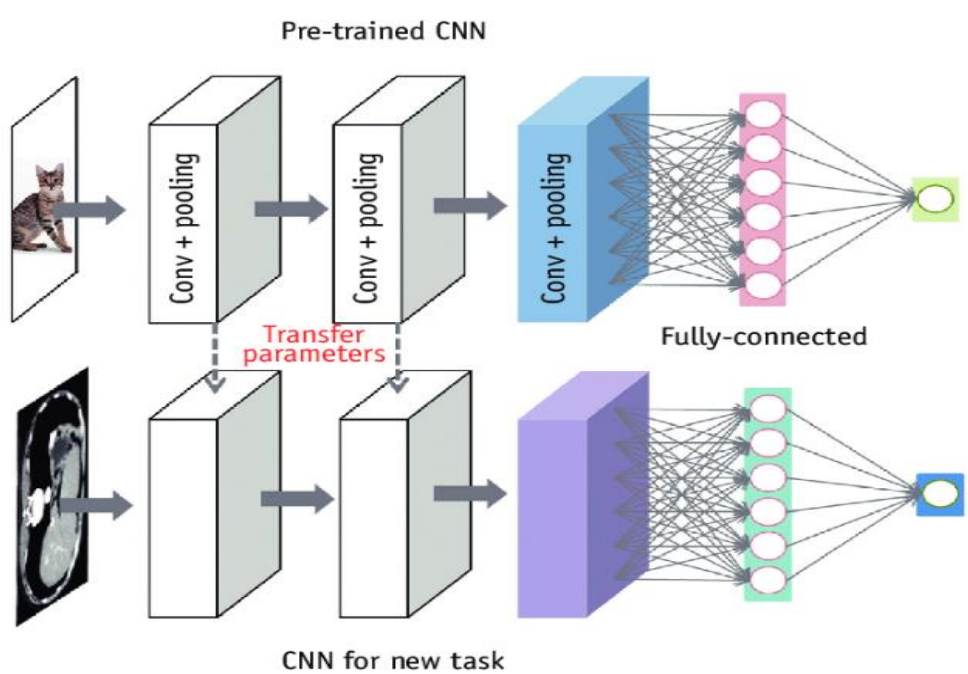
Transfer Learning-

A machine learning technique called transfer learning entails changing a model that was first trained on one task to make it appropriate for a different, related activity. The primary objective is to leverage the insights & knowledge acquired from addressing the initial task to enhance performance on the subsequent task, particularly when there is a scarcity of labeled data for the latter.

A model is frequently pre-trained on a large dataset designed for a particular job in the context of transfer learning, like image classification on ImageNet. This pre-trained model effectively captures meaningful features from the dataset on which it was originally trained.

The pre-trained model is adjusted or refined for the new goal rather than starting from scratch to train a new model for a target task. The fundamental idea is to apply the pre-training phase's knowledge to the new task, which works well in situations when there is a lack of labeled data for the second task.

Transfer learning is widely used in several fields, including as speech recognition, computer vision, & NLP. Notable pre-trained models such as VGG, BERT, & GPT have demonstrated considerable success in these domains, showcasing the effectiveness of transfer learning in real-world applications.



(iv) <u>Illustration of Transfer Learning</u>

CHAPTER-4

IMPLEMENTATION

Our project involved the complete implementation of a pipeline, covering tasks from data loading & preprocessing to model training & evaluation. The implementation details are outlined below:

Libraries used

torch: Pytorch is a library that provides high-level features like Tensor Computation & provides a framework for building deep learning models. It provides a flexible & dynamic computational graph, which is particularly useful for deep learning tasks.

torch.nn: This module in Pytorch provides classes & functions to build & train neural networks. It includes a variety of pre-built layers, optimization algorithms, etc.

NumPy: NumPy is a computing library for manipulation of large multi-dimensional arrays & matrices in Python.

Pandas: A widely used library for data manipulation & analysis & provides data structures like DataFrame & Series along with other functions for manipulating, cleaning, handling & analysing structured data.

Copy: A module providing a generic shallow & deep copy operations. It can be used for creating copies of objects.

matplotlib.pyplot: matplotlib.pyplot stands as a set of functionalities within the Matplotlib library, a widely used Python plotting tool. It serves as a user-friendly interface for crafting an array of static, animated, & interactive plots in the Python programming language.

torchvision.transforms: is a module available in the PyTorch-based **torchvision** library. It serves as a tool for performing image transformations & preprocessing in the context of computer vision tasks. The module encompasses a range of functions designed to enhance & manipulate images before they are input into a neural network. Common applications of this module involve augmenting data during the training of deep learning models & ensuring a standardized format for the input images.

Data Loading & Preprocessing: To enable seamless access to external data stored in Google Drive within the Google Colab environment, the Google Colab drive module is employed. After successfully mounting Google Drive, specific paths are designated for the training & testing datasets associated with Alzheimer's disease. These paths act as pointers to the locations where the datasets are stored within the mounted Google Drive. Following this, the image data is loaded using the **torchvision.datasets.ImageFolder** class. This class is well-suited for datasets organized in a manner where individual classes have their own subdirectories, housing images specific to each class. It is imperative to adjust the paths (**TRAIN_ROOT** & **TEST_ROOT**) to accurately reflect the real locations of the training & testing datasets within the Google Drive structure. Additionally, the **ImageFolder** class automatically assigns labels based on the subdirectory structure of the provided root paths.

Model Architecture Overview: A CNN architecture is implemented in PyTorch, encapsulated within a class named CNNModel. This class inherits from PyTorch's neural network module, denoted as N_N.Module. In the constructor (**init** method), the parent class is initialized through the use of the super function.

The code proceeds to instantiate the VGG16 pre-trained model from the torchvision.models module, explicitly specifying the utilization of weights pre-trained on the ImageNet dataset.

The number of input features for the final fully connected layer is computed & indicated as in feats, allowing the model to be customized for a specific classification task. The last fully connected layer of the VGG16 model is then swapped out with a brand-new linear layer that can output data for four classifications.

The forward method, responsible for the forward pass of the model, is defined. This method applies the VGG16 model to the input tensor & returns the resulting output.

Following the model definition, an instance of CNNModel is instantiated, & the model is relocated to a designated computing device, typically a GPU. This relocation is crucial for harnessing hardware acceleration during both the training & inference phases.

Training Procedure:-

Data Loaders Creation: Data loaders play a crucial role in managing data batches effectively during both the training & testing stages. In this particular project, two data loaders are established by leveraging PyTorch's torch.utils.data.DataLoader module. Specifically, the TRAIN_LOADER is crafted to load batches from the training dataset (TRAIN_DATA), while the TEST_LOADER is tailored for the test dataset (TEST_DATA). The training loader has been configured with a batch size of 32, enabling the simultaneous processing of 32 data samples in each iteration.

Training Setup: The subsequent section of the code establishes the foundational components required for the training process. The Cross-Entropy Loss (CROSS_ENTROPY_LOSS), a popular loss function for classification tasks, is defined in this as well. During training, the neural network model's (MODEL) parameters are updated via the Adam optimizer (OPTIMIZER). Furthermore, the training dataset will be processed ten times since the number of training epochs (EPOCHS) is set to 10.

Training Loop: The core of the training process is encapsulated within the training loop. The outer loop iterates through the defined number of training epochs (EPOCHS), while the inner loop traverses the batches within the training loader (TRAIN_LOADER). For each batch, the inputs & their corresponding labels are retrieved & transferred to the specified computing device (DEVICE). OPTIMIZER.zero_grad() is used to reset the optimizer's gradients to zero. The neural network model (MODEL) is then run through a forward pass, & the loss is calculated using the model's predictions & the actual labels. Following this, the backward pass is initiated (LOSS.backward()), & the optimizer proceeds to update the model parameters (OPTIMIZER.step()). Each batch's training loss is printed, offering information on the model's performance & the training process' convergence.

Inspect Predictions for the First Batch: After the conclusion of the training loop, the code advances to evaluate the model's performance on the initial batch of the test loader (TEST_LOADER). It extracts both inputs & labels from this batch, transfers the inputs to the designated computing device, & generates predictions utilizing the trained model. The model's accuracy on this specific batch is computed & displayed. Additionally, a Pandas DataFrame (COMPARISON) is instantiated to capture & store the actual labels (labels) alongside the predicted labels (outputs). This DataFrame serves as a valuable resource for subsequent analysis or visualization of the model's predictions.

LRP procedure:

1. `clone_layer` Function:

The `clone_layer` function serves the purpose of duplicating a specified neural network layer, allowing for the modification of its parameters through a user-defined function `G`. This function is versatile, handling adjustments to both the weight & bias parameters of the layer. Its significance lies in its pivotal role in the subsequent layerwise relevance propagation, facilitating the fine-tuning of layer parameters throughout the propagation process.

2. Layerwise Relevance Propagation:

Within this segment lies the integral logic for executing layerwise relevance propagation on the VGG16 model. The process entails systematically traversing the layers of the model, applying relevance propagation mechanisms at each step, & cumulatively aggregating relevance scores. This intricate procedure is central to uncovering the contributions of individual features within the model to the final predictions.

3. Calculate Relevances for a Specific Image:

This portion of the code undertakes the computation of relevance scores for a specific image, denoted by the variable `IMAGE_ID`, assumed to represent the 15th image in the dataset. The function `apply_lrp_on_vgg16` is employed to execute layerwise relevance propagation tailored to the VGG16 model for the given image. The resulting relevance scores are subsequently permuted & normalized, preparing them for visualization.

4. Get Predicted & Ground Truth Labels:

The code section responsible for retrieving the predicted label (`PRED_LABEL`) for the designated image plays a crucial role in understanding the model's classifications. This information is extracted using class indices obtained from the test data, shedding light on the model's predictions for evaluation purposes.

5. Check if the Image is Classified Correctly:

In the concluding part of the code, an assessment is made to ascertain whether the model correctly classified the specified image. The projected label & the actual ground truth label are compared to arrive at this conclusion. This evaluation step aids in validating the accuracy of the model's predictions.

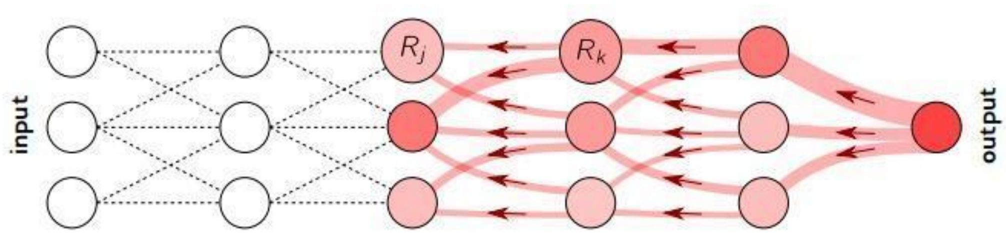


Illustration of the LRP procedure. Each neuron redistributes to the lower layer as much as it has received from the higher layer.

2. Illustration of the LRP

Ploting image relevances:

The following component is a fundamental element of a project focusing on the visualization of layerwise relevance propagation (LRP) outcomes from a VGG16 model. It serves the purpose of offering an intuitive representation of the model's decision-making process by showcasing relevances, the input image, & essential classification information.

The code begins by initializing a Matplotlib figure with two side-by-side subplots, each having dimensions of 10 by 5 inches. This ensures a clear & appropriately sized display for effective visualization.

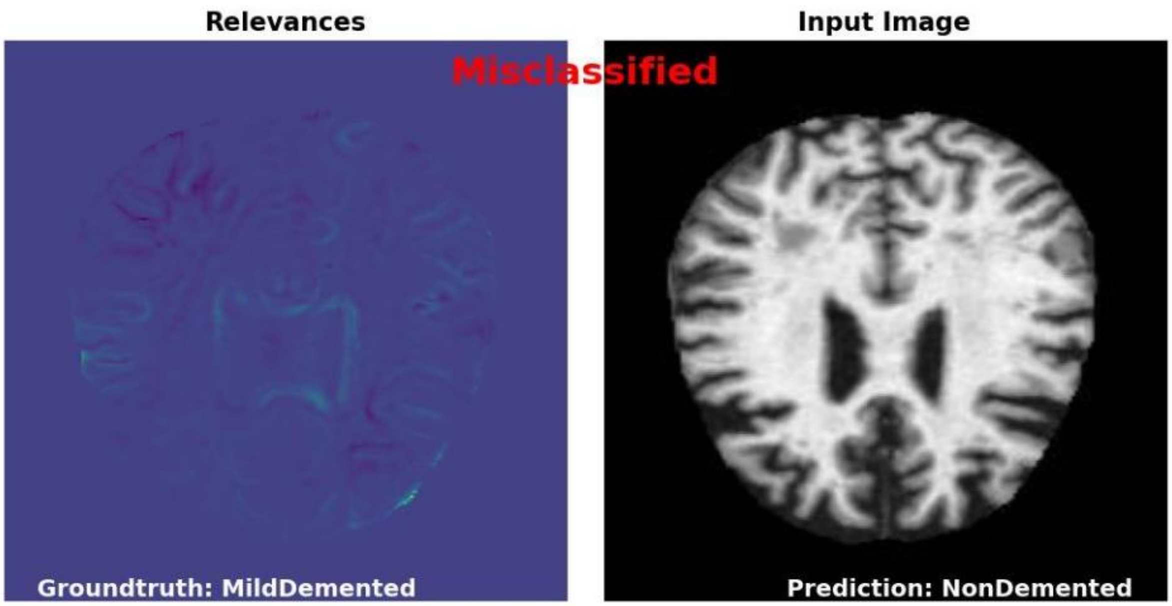
Subsequently, text annotations are strategically added to provide crucial information.

Annotations are positioned in the figure's bottom left & bottom right corners, respectively, to provide information about the ground truth label & the model's prediction label.. This contextual information enhances the interpretability of the visualized results.

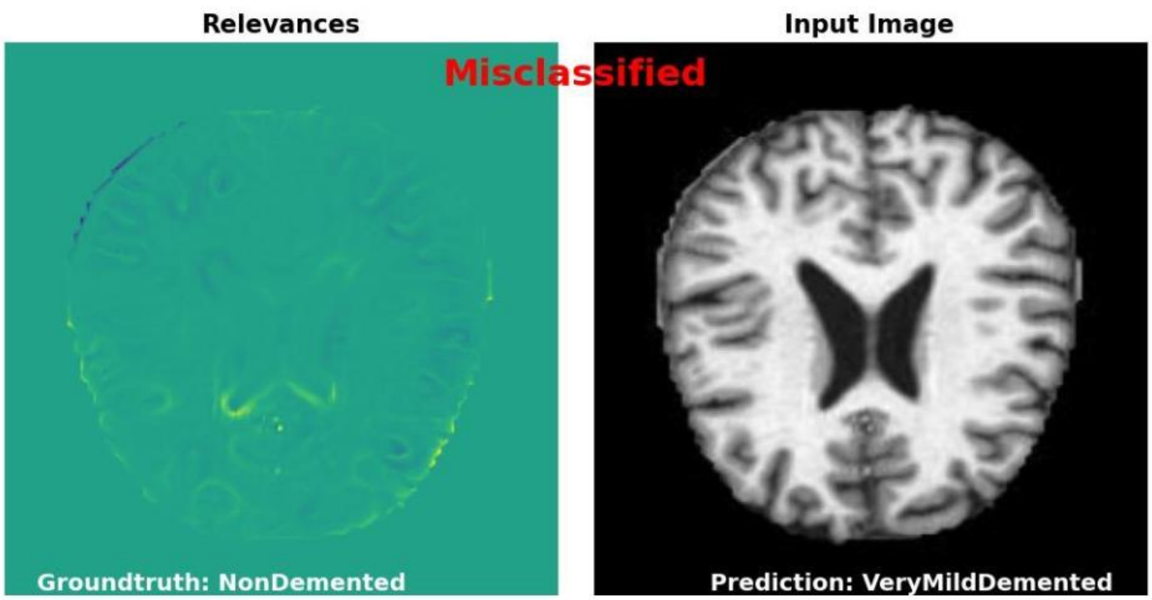
To further enhance the interpretability, a text annotation is positioned at the top center of the figure. This annotation serves as an indicator of the model's classification accuracy, specifying whether the model correctly classified or misclassified the given image. The use of green text color signifies correct classifications, while red text color is employed to denote misclassifications.

<u>Conclusion</u>

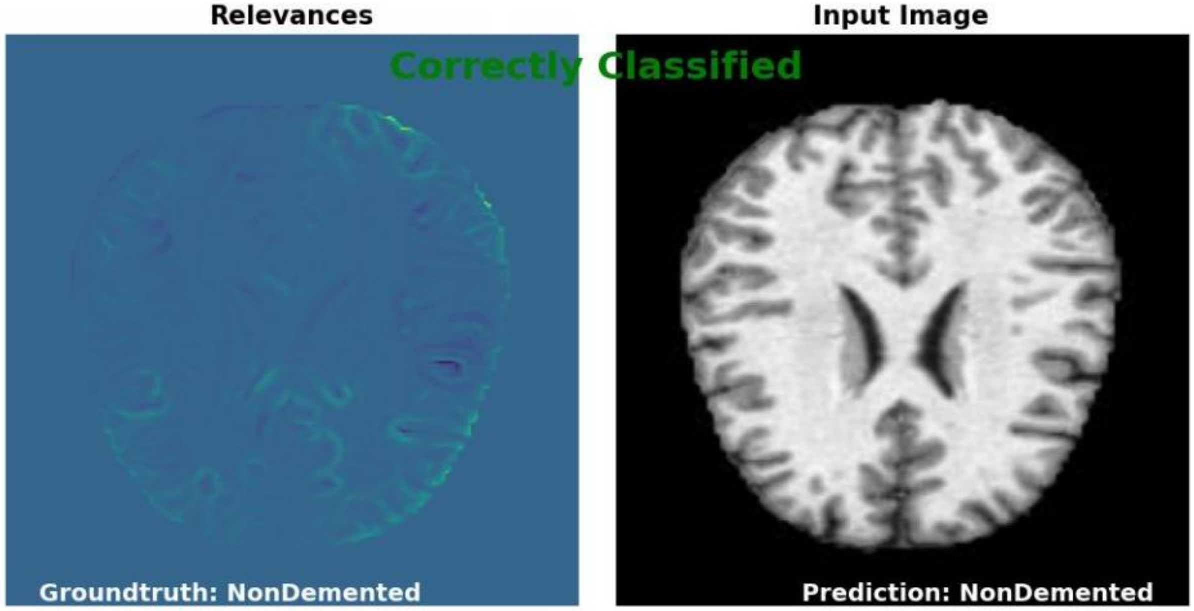
The model has shown batch accuracy ranging from 70% to 80%. We've achieved notable progress in recognizing accurate positives, accurate negatives, incorrect positives, & incorrect negatives by utilizing seismic graphs. Below there some of our output Images.



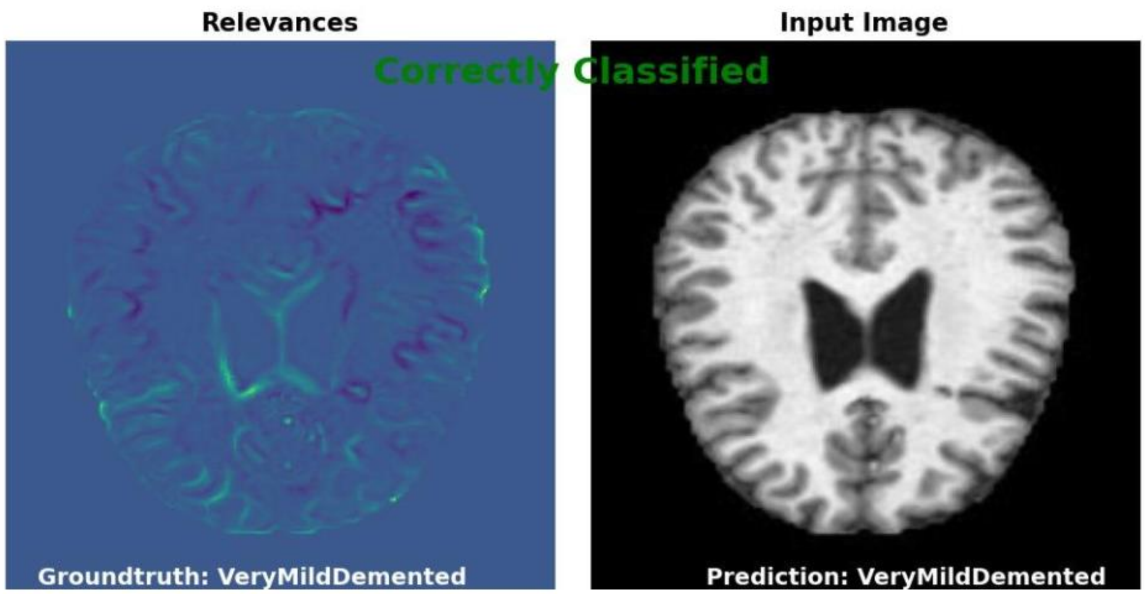
3. False negative



4. False positives



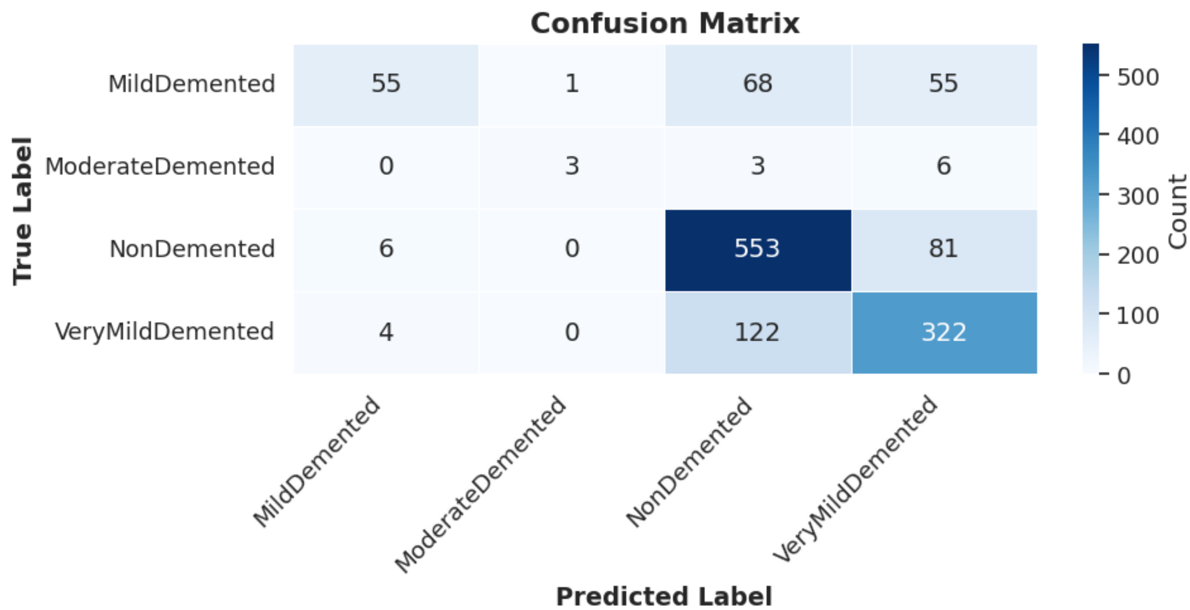
5. True Negative



6. True Positive

Confusion Matrix Explanation: The confusion matrix helps evaluate a classification model's performance. It breaks down predictions into four categories:

- Positive Correct (TP): Accurately predicted positive outcomes.
- Negative Correct (TN): Accurately predicted negative outcomes.
- Positive Incorrect (FP): Inaccurately predicted positive outcomes.
- Negative Incorrect (FN): Inaccurately predicted negative outcomes



Each column represents the anticipated class, while each row represents the actual class. The confusion matrix provides an overview of the model's strengths & weaknesses across various classes.

References	Modality	Data processing/training	Classifier	AD:NC acc.
Suk and Shen (2013)	MRI, PET, CSF	SAE	SVM	**95.9**
Liu et al. (2014)	MRI, PET	SAE + NN	Softmax	**87.76**
Suk et al. (2014)	MRI, PET	DBM	SVM	**95.35**
Li et al. (2014)	MRI, PET	3D CNN	Logistic regression	**92.87**
Li et al. (2015)	MRI, PET, CSF	RBM + drop out	SVM	**91.4**
Suk et al. (2015)	MRI, PET, CSF	SAE + sparse learning	SVM	**98.8**
Liu et al. (2015)	MRI, PET	SAE with zero-masking	Softmax	**91.4**
Cheng et al. (2017)	MRI	3D CNN	Softmax	**87.15**
Cheng and Liu (2017)	MRI, PET	3D CNN + 2D CNN	Softmax	**89.64**
Aderghal et al. (2017)	MRI	2D CNN	Softmax	**91.41**
Korolev et al. (2017)	MRI	3D CNN	Softmax	**80**
Vu et al. (2017)	MRI, PET	SAE + 3D CNN	Softmax	**91.14**
Liu et al. (2018a)	PET	RNN	Softmax	**91.2**
Liu et al. (2018b)	MRI	Landmark detection + 3D CNN	Softmax	**91.09**
Lu et al. (2018)	MRI, PET	DNN + NN	Softmax	**84.6**
Choi and Jin (2018)	PET	3D CNN	Softmax	**96**

SEN = TP/(TP + FN), SPE = TN/(TN + FP). TP, true positive; TN, true negative; FP, false positive; FN, false

References

[1]. Montavon, G., Binder, A., Lapuschkin, S., Samek, W., Müller, KR. (2019). Layer-Wise Relevance Propagation: An Overview. In: Samek, W., Montavon, G., Vedaldi, A., Hansen, L., Müller, KR. (eds) Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science(), vol 11700. Springer, Cham. https://doi.org/10.1007/978-3-030-28954-6_10.Link

[2]. Jo T, Nho K, Saykin AJ. Deep Learning in Alzheimer's Disease: Diagnostic Classification and Prognostic Prediction Using Neuroimaging Data. Front Aging Neurosci. 2019 Aug 20;11:220. doi: 10.3389/fnagi.2019.00220. PMID: 31481890; PMCID: PMC6710444. Link

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).Link

[4] Zhuang, Fuzhen, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. "A comprehensive survey on transfer learning." *Proceedings of the IEEE* 109, no. 1 (2020): 43-76.Link

[5] Gohel, Prashant, Priyanka Singh, and Manoranjan Mohanty. "Explainable AI: current status and future directions." *arXiv preprint arXiv:2107.07045* (2021).Link

[6] O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.Link

[7] Krishna, M & Neelima, M & Mane, Harshali & Matcha, Venu. (2018). Image classification using Deep learning. International Journal of Engineering & Technology. 7. 614. 10.14419/ijet.v7i2.7.10892.Link

Scope of Future Work-

In the future, we plan to focus on the following aspects to further improve our project:

Exploring the addition of layers to the model & testing different architectures.

Striving to achieve higher accuracy in Alzheimer's disease classification.

Conducting a time analysis to identify areas for optimization.

Investigating the feasibility of distributing the workload across multiple GPUs for faster training.

APPENDIX A:Abbreviations

Abbreviation	Meaning
ML	Machine Learning
LRP	Layer-wise Relevance Propagation
Eq	Equation
CNN	Convolutional Neural Network
AD	Alzheimer's Disease
AI	Artificial Intelligencer
CLF	Classification
DL	Deep Learning